OIPE
JAN 22 2008
PATENT & TRADEMARK OFFICE

AF

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of:	Attorney Docket No: ADAPP223
Mukund et al.	Examiner: TSAI, H.
Application No: 10/726,470	Group Art Unit: 2181
Filed: December 2, 2003	Date: September 15, 2006
For: NETWORKED PROCESSOR FOR A PIPELINE ARCHITECTURE	

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450 on January 15, 2008

Signed: _____
Michael L. Gencarella

TRANSMITTAL OF APPEAL BRIEF

Commissioner for Patents
Box: Board of Patent Appeals & Interferences
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is in furtherance of the Notice of Appeal filed in this case on March 27, 2007.

This application is on behalf of:

☐ Small Entity ☒ Large Entity

Pursuant to 37 CFR 1.17(f), the fee for filing the Appeal Brief is:

☐ $250.00 (Small Entity) ☒ $510.00 (Large Entity)

The proceedings herein are for a patent application and the provisions of 37 CFR 1.136 apply:

01/23/2008 SSITHIB1 00000135 10726470
460.00 OP
01 FC:1252

OIPE
JAN 22 2008
PATENT & TRADEMARK OFFICE
IAP95

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

EX PARTE Shridhar Mukund et al.

Application for Patent

Filed December 2, 2003

Application No. 10/726,470

FOR:

NETWORKED PROCESSOR FOR A PIPELINED ARCHITECTURE

APPEAL BRIEF

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail in an envelope addressed to: Commissioner for Patents, Alexandria, VA 22313-1450 on January 15, 2008

Signed: _____

Michael L. Gencarella

01/23/2008 SSITHIB1 00000135 10726470

02 FC:1402 510.00 OP
03 FC:1201 210.00 OP
04 FC:1202 1050.00 OP

MARTINE PENILLA & GENCARELLA, LLP
Attorneys for Applicants

01/23/2008 SSITHIB1 00000071 500850 10726470

02 FC:1402 510.00 DA

Void date: 01/23/2008 SSITHIB1 00000071 500850 10726470
01/23/2008 SSITHIB1 00000071 500850 10726470
02 FC:1402 510.00 CR

50mw

TABLE OF CONTENTS

Page No.

I. REAL PARTY IN INTEREST

The real party in interest is Adaptec, Inc., the assignee of the present application.

II. RELATED APPEALS AND INTERFERENCES

The Applicants are not aware of any related appeals or interferences.

III. STATUS OF CLAIMS

Claims 1-20 are pending in the subject application. Claims 1-20 have been rejected and are on appeal.

IV. STATUS OF AMENDMENTS

Applicants submitted an amendment of September 15, 2006, in response to a non-Final Office Action mailed on May 15, 2006. This amendment was the last entered amendment.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The subject invention is directed towards methods and apparatus for a processor configured to efficiently process incoming or outgoing packet data. The processor is arranged in a pipeline architecture, where one or more of the processors may be associated with a certain stage of the pipeline. The processor pipeline offloads previous processing performed by a central processing unit (CPU) of a host system, thereby freeing the CPU for other processing to improve system performance. The processor is configured to allow a single cycle access to a large address space.

Accordingly, a networking application processor as recited in claim 1 includes an input socket configured to receive data packets (see Figures 4 and 5, and page 9, lines 8-20) and a memory for storing instructions (See Figures 4 and 5 and page 12 lines 16-21). The processor includes circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle

access of an operand from a memory location (see Figures 6 and 7 and page 17, lines 15-25). An arithmetic logic unit (ALU) (Figures 6 and 8 and page 19, lines 5-10) and circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands (see Figure 6 and page 23, lines 10-21).

In another embodiment, a processor that includes an input socket configured to receive data packets (see Figures 4 and 5, and page 9, lines 8-20) and a memory for storing instructions (See Figures 4 and 5 and page 12 lines 16-21) is provided. The processor includes circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access from a memory location (see Figures 6 and 7 and page 17, lines 15-25). An arithmetic logic unit (ALU) (Figures 6 and 8 and page 19, lines 5-10) is also included. The ALU is configured to receive a first and a second operand where the second operand is specified from an internal register (see Figure 6 and page 18, lines 9-21) and the first operand has a mask enabling the ALU to process a non-masked segment of the first operand (see Figure 6 and page 21, lines 14-25).

In yet another embodiment, a processor capable of processing a data packet associated with a processing stage of a pipeline of processors is provided. The processor includes a data random access memory (RAM) configured to enable access to data structures (See Figures 5 and 6 and page 12, lines 14-20). Instruction fetch and decode circuitry configured to interpret instructions (See Figures 5 and 6 and pages 13-14) to be executed by an arithmetic logic unit (ALU) (Figures 6 and 8 and page 19, lines 5-10) is included. The instruction fetch and decode circuitry includes a read only

memory (ROM) (See Figures 5 and 6 and page 12, lines 14-20), the ROM configured to store code common to each processing stage associated with a pipeline of processors. The instruction and fetch decode circuitry includes a code RAM (See Figures 5 and 6 and page 12, lines 14-20) configured to download code specific to the processing stage and wherein the code specific to the processing stage is enabled for single cycle access. The processor includes instruction decode circuitry (See Figures 5 and 6 and pages 13-14) configured to recognize operating instructions. Execute and write back circuitry (See Figures 5 and 6 and page 12, lines 14-20 and page 13, lines 3-15) configured to set up operands to be processed by the ALU is included. The execute and write back circuitry includes internal registers (see Figure 6 and page 18, lines 9-21) for defining a first and a second operand. The processor further includes an arithmetic logic unit (Figures 6 and 8 and page 19, lines 5-10) for processing the first and second operands and align function circuitry (See Figures 6 and 8 and pages 22-23) for aligning the first and the second operands to be processed by the ALU. The align function circuitry causes the first and the second operands to be aligned by a lowest significant bit, wherein the align function circuitry supplies an extension to the each of the operands to allow the ALU to transparently process different size operands (see page 23 lines 3-20).

As referred to in this application, each of the pipelined processors include both hardware such as, input socket interface, star processor, output socket interface, hardware accelerator, and relevant software to access the hardware. (Figure 3 and related description on page 8, line 15 to page 11, line 5). For the pipelined processors the output socket interface of a first processor is in communication with an input socket interface of a second processor, and so on (see Figure 2).

It should be appreciated that the above description represents only a summary of the present invention. A more in-depth discussion of the present invention is provided in the Detailed Description section of the application.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Claims 1-6 and 14-20 were rejected under 35 USC 102(b) as being anticipated by Narayan et al. (U.S. Patent No. 5,822,559) (hereinafter "Narayan")

B. Claims 7-13 were rejected under 35 U.S.C. 103(a) as being unpatentable over Narayan et al. (U.S. Patent No. 5,822,559) (hereinafter "Narayan").

VII. ARGUMENT

A. Rejection of claims 1-6 and 14-20 under 35 U.S.C. § 102(b) as being unpatentable over Narayan.

Claims 1-6 and 14-20 were rejected under 35 USC 102(b) as being anticipated by Narayan.

Independent Claims 1 and 14

1. Narayan does not teach a single cycle access or circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands.

Narayan discloses a microprocessor including a plurality of decode units configured to detect double dispatch instructions and dispatch these instructions to decode units. The more complex instructions are executed by an MROM unit in a serial fashion. The complex instructions are detected prior to decode and dispatched to the MROM unit. The instruction alignment unit 206 is provided to channel variable byte length instructions from instruction cache 204 to fixed issue positions formed by decode units 208A-208D. Instruction alignment unit 206 independently and in parallel selects instructions from three groups of instruction bytes provided by instruction cache 204 and arranges these bytes into three groups of preliminary issue positions. Each group of the issue positions is associated with one of the three groups of instruction bytes. The preliminary issue positions are then merged together to form aligned issue positions, each of which is coupled to one of early decode units 207. Early decode units 207 are configured to realign the aligned issue positions created by instruction alignment unit 206 according to double dispatch instructions detected (see column 6, lines 35-65). Thus, the instructions alignment unit 206 arranges variable length instruction bytes to fixed issue positions. Narayan has nothing to do with pipelined processors and is only concerned with a single processor.

The Examiner asserts that Narayan discloses the feature of circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access of an operand from a memory location. In support of this assertion the Examiner refers to the main memory or the data cache of Figure 2. Applicants respectfully disagree with this assertion and submit that neither the Examiner nor Narayan specifies what the circuitry enables a single cycle access of an operand from memory. The Examiner states that the specification makes no specific definition of the term single cycle access. As mentioned in responses previously, page 17, lines 21-24 specifies that a single cycle access is where data is addressed and operated on

in a single clock cycle. The Examiner elects to ignore this statement and conveniently define a single cycle access as not being correlated with the clock cycle, which is in direct contradiction with the above mentioned portion of the specification. The Examiner then conclusorily states that fetching an instruction from memory is done in one clock cycle. Here again the Examiner ignores the single cycle access as being addressed and operated on in a single clock cycle. One skilled in the art will appreciate that fetching an instruction from memory will require an address to set up the fetch. The Examiner has simply ignored this feature to enable application of the reference as the reference is silent to addressing and operating on the instruction is a single clock cycle. The Examiner has not considered all of the features of the claims when analyzing the claims and the reference as nowhere in Narayan is this capability of a single cycle access of an operand discussed. Furthermore, as stated in column 66, lines 22-25 of Narayan, an extra clock cycle is needed for decoding instructions having more than 4 prefix bytes. Even if there was only one clock cycle used to decode the prefix byte, Narayan fails to specify this feature as the operand still needs to be operated on. The Examiner further asserts that the features of claim 2, that further define the single cycle access to enable the data to be addressed and operated on without being placed into a register is disclosed in column 10 the last paragraph of Narayan. There is no support whatsoever for this assertion in any of the paragraphs of column 10 of Narayan. Nowhere is it mentioned that data is addressed and operated on in a single clock cycle without being placed into a register.

The Examiner further states that the feature of circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands is taught in column 17, lines 32-42. Applicants respectfully disagree with the assertion that a prefix

designating the bit size of the operand discloses aligning operands by a lowest significant bit. As stated previously, the prefix deals with a single operand and nowhere is it mentioned that multiple operands are aligned by a lowest significant bit. The decoding process referred to by the Examiner is for determining the size of the operand to determine where to route the operand and for complex instructions (see column 16). Furthermore, nowhere in Narayan is it disclosed how the prefixes for multiple operands would be aligned. The Examiner responds that if any data is stored in a space larger that the data itself it must be aligned somehow and the lowest significant bit must be aligned at some point. Thus, for a 35 USC 102 rejection, the Examiner is conclusorily assuming that this feature is anticipated without relying on any particular portion of the cited reference or providing any technical basis for his assertion. Accordingly, claims 1-6 are allowable for at least these reasons.

Claim 14 includes the features of a read only memory (ROM), the ROM configured to store code common to each processing stage associated with a pipeline of processors and a code RAM configured to download code specific to the processing stage and wherein the code specific to the processing stage is enabled for single cycle access. The Examiner never provides any analysis of where Narayan teaches a ROM configured to store code common to each processing stage associated with a pipeline of processors. Narayan does not have a pipeline of processors and is silent as to this feature, and the Examiner has not provided any analysis of where Narayan teaches a pipeline of processors and instead simply ignores this feature. The Examiner's last response refers to column 22, lines 29-33 of Narayan for teaching a ROM, where it is stated that the instructions will be executed as micro-ROM sequences or fast path instructions. The Examiner then conclusorily states that the MROM stores micro instructions to be executed by the pipeline

without providing any reference where Narayan discloses a pipeline of processors as claimed.

In addition, as specified in claim 14, the code RAM and the ROM are part of the fetch and decode circuitry. The Examiner is asserting that a portion of the main memory and the MROM unit is included in the fetch and decode circuitry and refers to Figure 2 of Narayan. As illustrated in Figure 2, the MROM unit and the Main memory are different blocks and the main memory is not even in communication with the early decode units or the MROM. The Examiner broadly asserts that the MROM and main memory are part of the fetch and decode as they assist with the fetch and decode process. Applicants disagree with this broad assertion as under this logic anything remotely associated with the fetch and decode process is part of the fetch and decode circuitry. Further, the Examiner asserts that the main memory discloses both the DRAM configured to enable access to data structures and a ROM of the fetch and decode circuitry. Applicants respectfully disagree that the main memory can disclose both the ROM of the fetch and decode circuitry and the DRAM.

Claim 14 further includes the feature of align function circuitry for aligning the first and the second operands to be processed by the ALU, the align function circuitry the circuitry causing the first and the second operands to be aligned by a lowest significant bit, wherein the align function circuitry supplies an extension to the each of the operands to allow the ALU to transparently process different size operands. As discussed above, Narayan is silent as to aligning two operands and the Examiner has not provided any analysis on where multiple operands are aligned according to the lowest significant bits. In addition, the prefix codes of Narayan are for determining where to route the operands and not placing extensions on the operands. Claim 14 further includes the single cycle

access feature which is not disclosed by Narayan, as mentioned above. Accordingly, claims 14-20 are allowable for at least these reasons.

B. Rejections under 35 U.S.C. § 103:

Independent claim 7

2. Narayan does not teach circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access from a memory location.

Claims 7-13 were rejected under 35 U.S.C. § 103(a), as being unpatentable over Narayan. This rejection is respectfully traversed. Claim 7 includes the features of circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access from a memory location. As mentioned above, Narayan is incapable of enabling a single cycle access. The Examiner has provided no basis for this rejection besides an overly broad interpretation of the claim language by disregarding the specification. In addition the Examiner asserts that the feature of an arithmetic logic unit (ALU), the ALU configured to receive a first and a second operand; the second operand being specified from an internal register, the first operand having a mask enabling the ALU to process a non-masked segment of the first operand. The Applicants respectfully disagree with this characterization as the REGF cited by the Examiner is external to the functional units, i.e., the ALU. Claim 8 further defines the single cycle access and is patentable over Narayan for at least the above stated reasons with reference to claim 2.

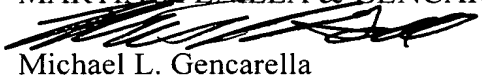
The Examiner has failed to meet the Examiner's burden of distinctly identifying each and every element of the claimed invention in the cited reference as Narayan does not disclose or teach each and every feature of the claims as discussed above.

The Applicants submit that the independent claims 1 and 14 are not anticipated under 35 USC 102(b) by Narayan and that claim 14 is patentable over Narayan.

C. Conclusion

In view of the foregoing reasons, the Applicants submit that each of the claims 1-20 are patentable. Therefore, the Applicants respectfully request that the Board of Patent Appeals and Interferences reverse the Examiner's rejections of the claims on appeal.

Respectfully submitted,
MARTINE PENILLA & GENCARELLA, LLP



Michael L. Gencarella
Reg. No. 44,703

710 Lakeway Drive, Suite 200
Sunnyvale, CA 94085
Direct Dial: 408.749.6905
Facsimile: (408) 749-6901
Customer Number 25920

OIPE IAP39 JAN 22 2008 PATENT & TRADEMARK OFFICE

VIII. CLAIMS APPENDIX

1. A networking application processor, comprising:

an input socket configured to receive data packets;

a memory for storing instructions;

circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access of an operand from a memory location;

an arithmetic logic unit (ALU); and

circuitry for aligning operands to be processed by the ALU, the circuitry for aligning operands causing the operands to be aligned by a lowest significant bit, wherein the circuitry for aligning the operands supplies an extension to the operands to allow the ALU to process different size operands.

2. The networking application processor of claim 1, wherein the instructions have a width of 96 bits, and wherein the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register.

3. The networking application processor of claim 1, wherein the different size operands are selected from the group consisting of 8 bit operands, 16 bit operands, and 32 bit operands.

4. The networking application processor of claim 1, further including:

an output socket for transmitting processed data; and

a 64 bit bus connecting the input socket and the output socket.

5. The networking application processor of claim 1, wherein the extension to the operand fills each higher bit with a value.

6. The networking application processor of claim 1, wherein the operand is selected from the group consisting of a source operand, a destination operand, an immediate operand, and an internal register operand.

7. A processor, comprising:

an input socket configured to receive data packets;

a memory for storing instructions;

circuitry configured to access data structures associated with a processing stage, the circuitry configured to access data structures enabling a single cycle access from a memory location; and

an arithmetic logic unit (ALU), the ALU configured to receive a first and a second operand; the second operand being specified from an internal register, the first operand having a mask enabling the ALU to process a non-masked segment of the first operand.

8. The processor of claim 7, wherein the instructions have a width of 96 bits, and wherein the single cycle access enables the data to be addressed and operated on in a single clock cycle without being placed into a register.

9. The processor of claim 7, wherein each of the instructions include a loadback feature enables random accesses to one of a source indirect register or a destination indirect register through indirect addressing.

10. The processor of claim 7, wherein the mask is associated with an immediate value of the first operand.

11. The processor of claim 7, wherein the first and the second operands are associated with a size selected from the group consisting of 8 bit operands, 16 bit operands, and 32 bit operands.

12. The processor of claim 7, wherein the first operand is selected from the group consisting of a source operand, a destination operand, an immediate operand, and an internal register operand.

13. The method of claim 7, wherein the memory location is a static random access memory (SRAM).

14. A processor capable of processing a data packet associated with a processing stage of a pipeline of processors, the processor comprising:
a data random access memory (RAM) configured to enable access to data structures;

instruction fetch and decode circuitry configured to interpret instructions to be executed by an arithmetic logic unit (ALU) , the instruction fetch and decode circuitry including,

a read only memory (ROM), the ROM configured to store code common to each processing stage associated with a pipeline of processors;

a code RAM, the code RAM configured to download code specific to the processing stage and wherein the code specific to the processing stage is enabled for single cycle access; and

instruction decode circuitry configured to recognize operating instructions;

execute and write back circuitry configured to set up operands to be processed by the ALU, the execute and write back circuitry including,

internal registers for defining a first and a second operand;

an arithmetic logic unit for processing the first and second operands; and

align function circuitry for aligning the first and the second operands to be processed by the ALU, the align function circuitry the circuitry causing the first and the second operands to be aligned by a lowest significant bit, wherein the align function circuitry supplies an extension to the each of the operands to allow the ALU to transparently process different size operands.

15. The processor of claim 14, wherein the extension to each of the operands fills each higher bit with a value.

16. The processor of claim 14, wherein the different size operands have a width selected from the group consisting of 8 bits, 16 bits, and 32 bits.

17. The processor of claim 14, wherein the operating instructions wherein the operating instructions are formatted as 96 bit instructions, each of the 96 bit instructions including a single return bit.

18. The processor of claim 14, wherein the processor is configured as a two stage pipeline for pipelining an instruction fetch and decode operation and an execute and write back operation.

19. The processor of claim 14, wherein the operating instructions include microcode configured to predict a likely direction for a branch instruction.

20. The processor of claim 19, wherein no operation (NOP's) instructions are included, the NOP's configured block an invalidated pre-fetched instruction.

IX. EVIDENCE APPENDIX

There is currently no evidence entered and relied upon in this Appeal.

X. RELATED PROCEEDINGS APPENDIX

There are currently no decisions rendered by a court or the Board in any proceeding identified in the Related Appeals and Interferences section.